IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 28, 2018, accepted June 29, 2018, date of publication July 5, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2853132

Scalable Remote Rendering Using Synthesized Image Quality Assessment

XIAOCHUAN WANG[1], XIAOHUI LIANG[1], (Member, IEEE), BAILIN YANG[2], AND FREDERICK W. B. LI[3]

[1]State Key Laboratory of Virtual Reality Technology and System, Beihang University, Beijing 100191, China
[2]School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
[3]Department of Computer Science, Durham University, Durham DH1 3LE, U.K.

Corresponding author: Xiaohui Liang (liang_xiaohui@buaa.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1002702, in part by the National Nature Science Foundation of China under Grant 61572058, and in part by the National Natural Science Foundation of China under Grant 61472363.

ABSTRACT Depth-image-based rendering is widely used to support 3-D interactive graphics on low-end mobile devices. Although it reduces the rendering cost on a mobile device, it essentially turns such a cost into depth image transmission cost or bandwidth consumption, inducing performance bottleneck to a remote rendering system. To address this problem, we design a scalable remote rendering framework based on synthesized image quality assessment. Especially, we design an efficient synthesized image quality metric based on just noticeable distortion (JND), properly measuring human-perceived geometric distortions in synthesized images. Based on this, we predict quality-aware reference viewpoints, with viewpoint intervals optimized by the JND-based metric. An adaptive transmission scheme is also developed to control depth image transmission based on perceived quality and network bandwidth availability. Experimental results show that our approach effectively reduces the transmission frequency and the network bandwidth consumption with perceived quality on mobile devices maintained. A prototype system is implemented to demonstrate the scalability of our proposed framework to multiple clients.

INDEX TERMS Depth-image-based rendering, remote rendering, synthesized image quality assessment, system scalability, transmission scheme.

I. INTRODUCTION

With the advances in mobile devices and wireless transmission technologies, remote rendering using *depth-image-based rendering* (DIBR) becomes popular for supporting interactive 3D graphics on low-end mobile devices. Good examples include 3D model display [1]–[3], volume data visualization [4], and 3D scene navigation or virtual environment walkthrough [5], [6]. Instead of sending clients explicit geometric data for local rendering, DIBR-based remote rendering occasionally sends clients reference depth images, where each comprises a texture image of a rendered view and its associated depth map, forming inputs for a client to synthesize required virtual views through 3D warping [1] without collecting such views from the server continuously. This significantly reduces rendering cost and storage consumption at low-end mobile devices, while supporting flexible and interactive user interactions.

Typical DIBR-based remote rendering framework has three main components, namely *depth image compression/decompression*, *depth image transmission* and *virtual view synthesis*. The focus of depth image compression/decompression is to reduce redundancy from depth map to ensure coding efficiency [7]–[15]. Virtual view synthesis uses pre-received reference depth images to generate proper virtual views (synthesized images) by minimizing geometric distortions, particularly holes around disoccluded regions [16]–[23]. Depth image transmission predicts an optimal number of reference viewpoints based on user interaction and sends users corresponding reference depth images. However, due to transmission cost, system scalability becomes problematic when supporting multiple clients.

Current depth image transmission techniques can be categorized into two types. One is *time-interval-based*, which transmits reference depth image with a fixed time interval [1], [5], [6] by predicting reference viewpoints according to the velocity of user viewpoint movement, thereby inducing an excessively high transmission frequency. The other one is *content-based*, where reference viewpoints

are pre-determined by pixel errors of synthesized images [3]. No extra depth image transmission is required unless user viewpoint moves outside the interval of the current reference viewpoint. Despite the method can reduce transmission frequency, the Mean Squared Error (MSE) metric used in determining viewpoint interval may under-estimate the human perceived quality of a synthesized image, as unnoticeable geometric distortions are not taken into account. This leads to non-optimal viewpoint interval prediction. The method also does not support network bandwidth adaptation, which is important for serving clients with varying network bandwidth.

To effectively reduce transmission cost, we design a novel depth image transmission strategy by enlarging the viewpoint interval with respect to the perceived quality of synthesized images. A Just Noticeable Distortion (JND) based synthesized image quality assessment is proposed to properly measure geometric distortions in a synthesized image. According to assessment results, reference viewpoints are predicted with an optimized viewpoint interval. Particularly, we predict reference viewpoints in a multi-scale way to adapt different available network bandwidth. Also, an adaptive transmission scheme is integrated to fetch a reference viewpoint from the predicted reference viewpoint set according to user interaction. The transmission scheme additionally accounts for the rendering resolution of a depth image, further reducing rendering and bandwidth consumption. Our major contributions are as follows:

- We propose an efficient synthesized image quality metric based on JND cues, where the geometric distortions in synthesized image are properly measured with respect to human perception.
- We predict a multi-scale reference viewpoint set under different bandwidth constraints, where the viewpoint interval of reference viewpoint is determined by our proposed JND-based metric.
- An adaptive transmission scheme is proposed to synthetically determine customized transmission timing and rendering resolution of reference depth image for each client.
- We implement a prototype system by integrating our scalable remote rendering framework, demonstrating system scalability toward multiple clients through simulations with interactive 3D graphics scenarios.

The rest of this paper is organized as follows. Related works are reviewed in Section II. Section III presents our design of the JND-based synthesized image quality metric. Our proposed scalable remote rendering framework is depicted in Section IV. Experimental results and prototype system evaluation are provided in Section V and Section VI, respectively. Finally, Section VII concludes our work.

II. RELATED WORK

Remote rendering framework can be categorized into model-based rendering (MBR) [24] and image-based rendering (IBR) [25]. With the increase in geometry complexity, MBR becomes very challenging for a client to carry out rendering interactively, particularly for low-end mobile devices. In contrast, IBR relies the server to perform rendering and send a client rendered views with optional auxiliary information. This essentially trades image data transmission for reduction in resource consumption at clients, making interactive 3D graphics be possible to low-end mobile devices.

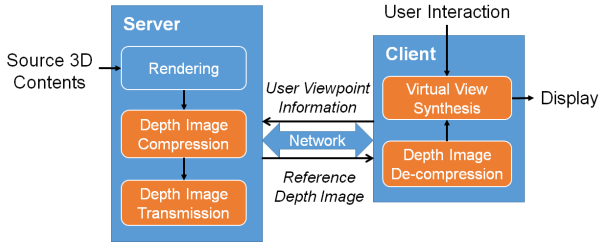

FIGURE 1. Basic workflow of typical DIBR-based remote rendering.

Derived from IBR, many DIBR-based remote rendering frameworks have been proposed [1], [3]–[6]. As illustrated in Fig. 1, a typical DIBR-based remote rendering framework comprises depth image compression/de-compression, depth image transmission and virtual view synthesis. Our proposed work focuses on optimizing depth image transmission, while many research efforts in the literature were on depth image compression/de-compression [8]–[15] and virtual view synthesis [16]–[23], [26]. All these works do not impose explicit control on depth image transmission. Our work is rather complementary to [3] since we focus on optimizing the reference viewpoint prediction, learning to optimal reference depth image transmission. In the following, we only discuss most relevant existing works.

Current depth image transmission strategies can be categorized into time-interval and content-based transmission:

A. TIME-INTERVAL-BASED TRANSMISSION

It transmits reference depth images by predicting reference viewpoint with a pre-defined time interval according to user interaction [1], [4]–[6]. Since a predicted reference viewpoint may easily deviate from the actual user viewpoint movement, inducing serious distortions to synthesized image, high-frequency reference viewpoint prediction is necessary. For instance, [1] set a fixed time interval of 200ms for prediction. Bao and Gourlay [5], [6] inherited this transmission strategy and additionally reduced the amount of data transmission by sending image differences between reference depth images instead of the original forms of those images. Zellmann *et al.* [4] also maintain a fixed time interval for reference viewpoint prediction, but resorting efficient organization of depth information on top to facilitate volume rendering. Frequent reference viewpoint prediction with fixed time intervals can effectively support arbitrary user interaction as the predicted reference viewpoints can usually well match user viewpoint movements. However, the induced

rendering cost and network bandwidth consumption become critical concerns.

B. CONTENT-BASED TRANSMISSION

Content-based transmission only transmits reference depth images when the synthesized image quality becomes unacceptable. Shi *et al.* [3] proposed a reference viewpoint prediction based on pixel errors of synthesized image. Transmission occurs when user viewpoint moves outside the coverage of current reference viewpoint. Transmission frequency is thereby relied on the predicted viewpoint interval. Comparing with time-interval-based methods, it effectively reduces redundant depth image transmission. Similar reference viewpoint prediction is also adopted in other IBR framework [30], [31].

Despite content-based transmission is meant to reduce transmission frequency, saving rendering cost and network bandwidth consumption, a proper image quality metric is required to facilitate this. Recent work [3] resorted image pixel errors, which are measured by MSE, to support reference viewpoint prediction. It under-estimated synthesized image quality as unnoticeable geometric distortions were not considered. In addition, most existing works were not scalable since they did not handle the increase in network bandwidth consumption causing by multiple concurrent clients.

Despite synthesized image quality assessment is important for reference viewpoint prediction, current 2D image quality metrics [32]–[35], including MSE, cannot properly measure geometric distortions in synthesized image toward human perception [36]. Recently, novel synthesized image quality metrics are proposed [36]–[41]. However, most of them are time-consuming, prohibiting real-time remote rendering.

III. JND-BASED SYNTHESIZED IMAGE QUALITY ASSESSMENT

Effectiveness of synthesized image quality assessment is critical to depth image transmission efficiency (See Section IV-A), because this affects reference viewpoint prediction as discussed in Section II. This section depicts how we derive such a metric. We first discuss geometric distortions in DIBR synthesized images. We then elaborate the design of a full-reference (FR) synthesized image quality metric based on JND cues. We finally present the no-reference (NR) version of our JND-based metric and the computation complexity. Table 1 summarizes the main notations used in the following.

Given a reference viewpoint v_{ref}, its associated depth image $< I(v_{ref}), D(v_{ref}) >$ is obtained by directly rendering from the source 3D scene. The synthesized virtual view $< I'(u), D'(u) >$ of a current user viewpoint u can then be generated by 3D warping using the reference depth image:

$$< I'(u), D'(u) > = warping(< I(v_{ref}), D(v_{ref}) >, v_{ref} \rightarrow u) \quad (1)$$

where the synthesized image $I'(u)$ is perceived by the user. Details of 3D warping can be found in [1] and [26].

TABLE 1. Main notations.

Symbol	Definition
$V, v \in V$	set of reference viewpoints and specific reference viewpoint v, respectively
$P, p \in P$	set of possible paths of user viewpoint movement and specific path p, respectively
$u, \vec{v}$	user viewpoint and associated velocity, respectively
$< I, D >$	depth image composed with a texture image I and the associated depth map D, respectively
$< I', D' >$	DIBR synthesized image and the associated warped depth map, respectively
Q	objective quality of synthesized image
H	viewpoint coverage in terms of scene content
$d, \Delta d$	viewpoint interval and unit distance step between two adjacent viewpoints, respectively
BW	available network bandwidth
res	image resolution

(a) (b)

FIGURE 2. Example of a synthesized image and its undistorted version. (a) undistorted image, (b) Synthesized image.

Fig. 2(b) shows an example of a synthesized image. By comparing with its undistorted version (Fig. 2(a)), geometric distortions, like holes and cracks, are found in various regions of the image, highlighted by color boxes. As illustrated in Fig. 2(b), geometric distortions are sensitive to human perception when they are located around bright regions (highlighted in red box) but can hardly be observed when locating at dark regions (highlighted in blue box). Besides, distortions around simple texture regions (highlighted in yellow box) are much easily perceived than that around complex regions (highlighted in green box). These refer as non-structure and locality of geometric distortions. Such local distortions cannot be correctly quantified by conventional 2D image metrics, such as MSE, which only globally measures image pixel errors [36].

To address the problem, we incorporate JND cues into synthesized image quality assessment. JND leverages biological and physiological cues to measure human sensitivity to visual signal difference [42]. It helps quantify local pixel errors under different types of surroundings (image sub-regions) based on human perception, while MSE only provides a global measure of image distortion. In a JND-based metric, the amount of distortions are quantized into JND levels, with each representing a certain level of acceptable distortion [43].

According to observations, we choose local luminance adaptation and local texture contrast masking [42] for synthesized image quality assessment. The per-pixel JND level is formulated as follows:

$$JND(x,y) = LA(x,y) + CM(x,y) - \lambda \times \min\{LA(x,y), CM(x,y)\}, \quad (2)$$

where LA and CM denote local luminance adaptation and local texture contrast masking, respectively. (x,y) denotes a pixel position, and λ is a constant indicating competition effect between luminance adaptation and texture contrast masking. A larger λ represents more significant overlapping effect. In our work, we experimentally set $\lambda = 0.3$. The local luminance adaptation is formulated as follows:

$$LA(x,y) = \begin{cases} 17 \times (1 - \sqrt{\frac{\bar{I}(x,y)}{127}}) + 3, & \bar{I}(x,y) \le 127 \\ \frac{3}{128} \times (\bar{I}(x,y) - 127) + 3, & otherwise \end{cases} \quad (3)$$

In Eq. (3), $\bar{I}(x,y)$ computes the mean intensity of the 5×5 local region centered at (x,y). The local texture contrast masking is denoted as:

$$CM(x,y) = G(x,y)W(x,y) \quad (4)$$

where $G(x,y)$ is the mean gradient of the same 5×5 region:

$$G(x,y) = \max_{k=1,2,3,4}\{grad_k(x,y)\} \quad (5)$$

with

$$grad_k(x,y) = \frac{1}{16}\sum_{i=-2}^{2}\sum_{j=-2}^{2} I(x+i, y+j)g_k(i,j), \quad (6)$$

where $g_k(i,j)$ are four directional high-pass filters for texture contrast detection. More details can be found in [44] and [45].

Specifically, we use $W(x,y)$ to measure the locality effect of geometric distortions. Different weights are assigned to image edge pixels according to their surroundings:

$$W(x,y) = \begin{cases} 0.1, & (x,y) \in \Omega_{dis}, \\ 0.3, & (x,y) \in \Omega_{con}, \\ 1.0, & otherwise. \end{cases} \quad (7)$$

where Ω_{dis} indicates pixels belonging to the edges of disocclusion regions, and Ω_{con} denotes other edges. The edges of disocclusion regions are detected from the depth map, i.e., depth discontinuous in the depth map. While other edges are detected from the synthesized image. It fits for the observation that edges belonging to disocclusion regions have low local texture contrast masking effect, thereby being easily noticeable in general.

We design a FR synthesized image quality metric, which requires an undistorted image generated directly from rendering the 3D content at a user viewpoint $I(u)$ as the ground truth. Quality evaluation of a synthesized image $I'(u)$ is started with calculating per-pixel JND levels through Eq. 2. We then compare the difference between $I(u)$ and $I'(u)$, judging whether each pixel error falls below the corresponding JND level. It is intuitive that a pixel differing from its ground truth with a distortion smaller than the corresponding JND level, such distortion is not noticeable to human perception. We then mark those pixels with 1, and mark others with 0:

$$f(x,y) = \begin{cases} 1, & if \ \ |I'(x,y) - I(x,y)| \le JND(x,y), \\ 0, & otherwise. \end{cases} \quad (8)$$

Finally, we utilize the ratio of pixels being below their associated JND levels, to measure the perceived quality of a synthesized image:

$$Q_{syn}(I'(u)) = \frac{\sum_1^m \sum_1^n f(x,y)}{m \times n}, \quad (9)$$

where $m \times n$ denotes the image size.

Comparing with MSE, our metric matches better with human perception. First, our metric has a limited scale ranging from 0 to 1, representing a range from worst to best perceived quality. However, MSE ranges from 0 to $+\infty$, hardly representing quantifiable human subjective scores. Second, our metric indicates the user perception level monotonously, while MSE may yield different values to the same perception level [32]. Finally, our metric properly measures distortions against their surroundings, while MSE is a global measurement being insensitive to local distortions.

(a) (b)

FIGURE 3. The differential map depicts pixel errors of Fig. 2(b) against Fig. 2(a), and the JND map calculated with Eq. (2) contains high intensity pixels denoting noticeable distortions. (a) Diffrential map. (b) JND map.

Fig. 3(a) shows the differential map of Fig. 2(b) against Fig. 2(a), while Fig. 3(b) depicts the JND map being able to highlight pixels exceeding their JND levels. The JND map can well reflect how human perceive local distortions, while the differential map comprises too much signals which do not contribute to human perceived distortions but accumulated pixel errors.

We also design a NR JND-based metric for the client-side at which the undistorted ground truth image of the current view is unlikely available. The metric is constructed as above but replacing undistorted ground truth image with reference texture image. To further reduce user interaction latency, we implement our metrics on GPUs, where per-pixel operations are handled with *OpenGL Shader*.

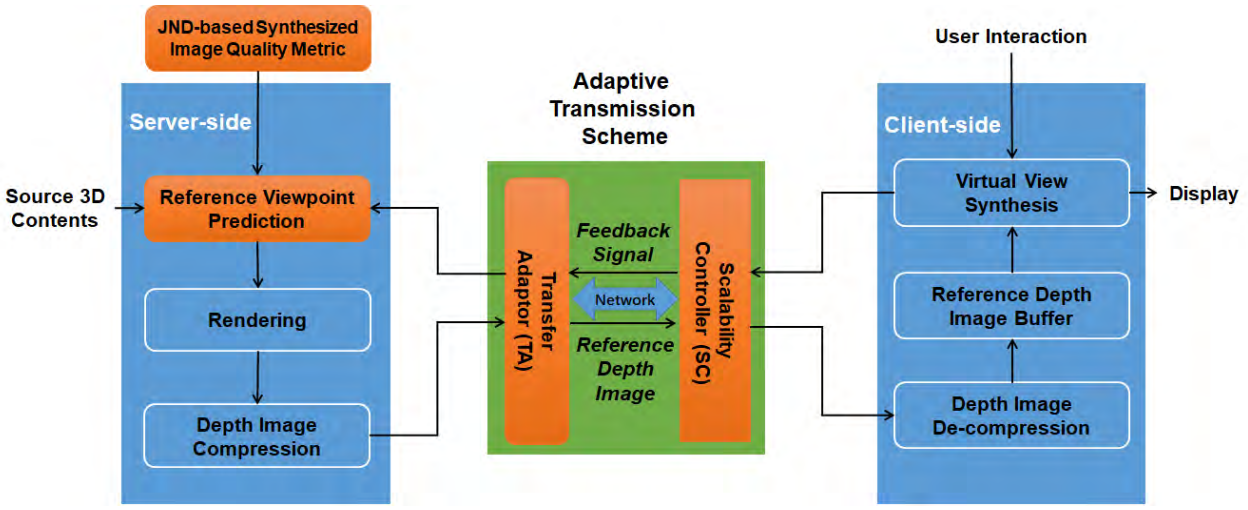

FIGURE 4. Overview of our proposed remote rendering framework, with our contributions highlighting in orange boxes.

IV. SCALABLE REMOTE RENDERING FRAMEWORK

With the proposed JND-based synthesized image quality metric, we propose a scalable remote rendering framework for 3D interactive graphics on mobile devices, as illustrated in Fig. 4. In this section, we first describe the proposed framework, and then present our quality-aware reference viewpoint prediction and adaptive transmission scheme accordingly.

The primary task of the server-side is to perform the reference viewpoint prediction based on our proposed JND-based synthesized image quality metric, generating a multi-scale reference viewpoint set accordingly to support bandwidth adaptation (see Section IV-A). The rendering and transmission of reference depth images are controlled by *Transfer Adaptor* (TA), which is part of the proposed adaptive transmission scheme (see Section IV-B). It receives feedback information from the client-side and fetches a proper reference viewpoint from the constructed reference viewpoint set. It also decides the timing for depth image transmission. Another server-side task is depth image compression, where the texture image and depth map are separately compressed.

For the client-side, user interaction is transformed into user viewpoint movement, which is synchronized to the server-side through *Scalability Controller* (SC) as part of the adaptive transmission scheme (see Section IV-B). SC also monitors the perceived quality of a synthesized image on mobile device as well as available network bandwidth, with which to dynamically negotiate with TA as feedback signal about the required scale of reference viewpoint set and rendering resolution of next reference depth image. When the required reference depth image is received, the corresponding texture image and depth map are separately decompressed, caching into a buffer. A relevant virtual view can then be synthesized by 3D warping.

A. REFERENCE VIEWPOINT PREDICTION

Suppose the parameters, e.g., field of view, of all reference viewpoints are uniform. The key factor affecting geometric distortions is the coverage of a reference viewpoint, which can be transformed into the viewpoint interval between two adjacent reference viewpoints. Fig. 5 illustrates the coverage of reference viewpoints. As shown in the figure, v_1 and v_2 provide an overlapped viewpoint coverage $H(v_1) \cup H(v_2)$. Any synthesized image in between can be reconstructed with the two reference viewpoints without quality degradation. However, v_1' and v_2' separate quite apart, and that cannot provide complete information to properly reconstruct synthesized images between them. For instance, the synthesized image at u as in Fig. 5 contains disocclusion regions from the two reference viewpoints, inducing geometric distortions. Note that the perceived quality is related to the viewpoint interval, e.g., d or d'.

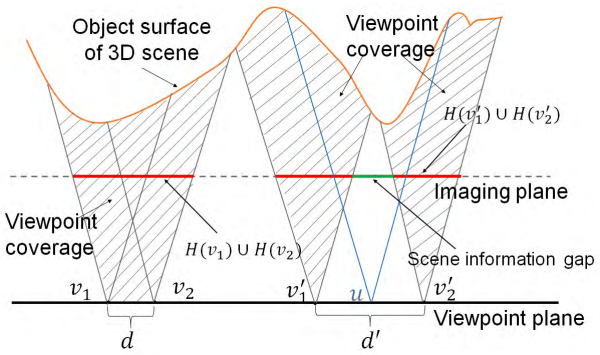

FIGURE 5. Illustration of viewpoint coverage: Reference viewpoints v_1 and v_2 are located very close with viewpoint coverage overlapped, while a significant gap is found between the coverage of v_1' and v_2'. Synthesized image at u cannot be properly generated without quality degradation due to the increase in viewpoint interval.

Maximizing such an interval sacrifices acceptable quality but reducing depth image transmission when user viewpoint moves within the interval. Shi *et al.* [3] decides the viewpoint interval based on MSE of synthesized image. The predicted reference viewpoint is hence non-optimal,

since MSE cannot properly reflect human perception, and usually under-estimates the perceived quality of a synthesized image. In contrast, we determine viewpoint intervals with our proposed JND-based metric, which can properly measure geometric distortions as described in Section III. Essentially, our metric can tolerate pixel errors below a required JND level, generating a sparser reference viewpoint set.

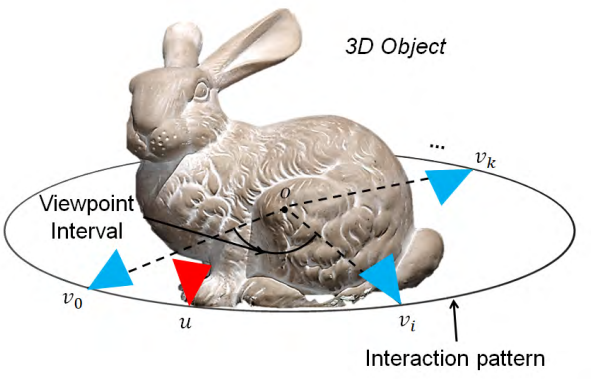

FIGURE 6. Illustration of predicted reference viewpoint set: Red triangle simulates user viewpoint moving along a circular orbit. Blue triangles depict reference viewpoints.

As shown in Fig. 6, the viewpoint interval between two adjacent viewpoints v_0 and v_i can logically be quantified by $i \times \Delta d$, where Δd denotes the unit displacement of user interaction, e.g. the unit length for viewpoint translation, or the unit angle for viewpoint rotation, and i denotes the number of interaction steps involved. A synthesized image of u with v_0 as the reference viewpoint, is generated by:

$$I'(u) = warping(< I(v_0), D(v_0) >, v_0 \rightarrow u), \qquad (10)$$

and the optimal viewpoint interval can be formulated as:

$$d = \arg\max_{d} |Q_{syn}(I'(u)) - Q_{JND}| \qquad (11)$$

where $Q_{syn}(I'(u))$ is the synthesized image quality, and Q_{JND} is a preset quality threshold, both of them are related to the proposed JND-based metric. If a lower quality threshold is set, more image pixels will exceed the JND level and that a larger viewpoint intervals view will be resulted.

To solve Eq. (11), a naive way is to explore all possible viewpoints in a 3D scene measuring the quality of each synthesized image and deducing the result. However, this is extremely computationally expensive and is not favorable to perform during runtime. We instead pre-compute the reference viewpoint set in polynomial time. To allow feasible implementation, we confine the reference viewpoint prediction to only all possible paths of user viewpoint movement P. For each path, we also perform a discrete number of operations instead of infinite ones.

Accordingly, we propose a full-search algorithm to construct the reference viewpoint set, as listed in Algorithm 1. The search starts with a randomly selected viewpoint v_0. For each candidate viewpoint $v_{i\times\Delta d}$, we warp v_0 to it,

Algorithm 1 Full-Search Algorithm for Reference Viewpoint Prediction

Require: $v_0, Q_{JND}, \Delta d, d_{max}, P, V$
  for $p \in P$ **do**
    $v_{ref} = v_0, V = \Phi$
    for $|v_{ref} - v_0| \leq d_{MAX}$ **do**
      $i = 1$
      if $Q_{syn}(I'(v_{i\times\Delta d})) \geq Q_{JND}$ **then**
        Warping $I'(v_{i\times\Delta d})$ with v_{ref}
        $i++$
      end if
      $v_{ref} = v_{i\times\Delta d}$ with $d = i \times \Delta d$
      $V = V \cup v_{ref}$
      $v_0 = v_{ref}$
    end for
  end for

and then evaluate the perceived quality of $I'(v_{i\times\Delta d})$ with proposed JND-based metric, judging whether the perceived quality is below the preset quality threshold Q_{JND}. The search terminates with two conditions. One is finding the optimal viewpoint interval $i \times \Delta d$ where the synthesized image quality is falling below the preset quality threshold. We then select $v_{i\times\Delta d}$ a new reference viewpoint and repeat the search accordingly. The other one is that the search reaches a maximal distance d_{MAX} according to the 3D scene boundary. A reference viewpoint set V is constructed when all possible paths are searched.

To make our method scalable to different network bandwidth conditions, we extend the reference viewpoint set to a multi-scale one, allowing various scales to be constructed based on different available network bandwidth requirements. Such that in each scale, viewpoint intervals are determined by different quality thresholds. In practice, three quality thresholds, i.e., $Q_{JND} = 0.990, 0.985, 0.980$, are used. The predicted reference viewpoints become sparser when quality threshold decreases. Consequently, less reference depth images are required to transmit, providing network bandwidth adaptation.

The computational complexity of the proposed full-search algorithm is no larger than $O(d_{max}|P|)$, where $|P|$ is the number of possible paths. As the value of $|P|$ increases, the complexity of the full-search algorithm becomes larger but the flexibility of user interaction improves. In addition, the value of d_{max} probably increases with the size of a 3D scenes. The unit distance Δd in this case can be resized to maintain the time efficiency.

B. ADAPTIVE TRANSMISSION SCHEME

With the predicted multi-scale reference viewpoint set, we design an adaptive transmission scheme for real-time DIBR-based remote rendering. Our depth image transmission strategy is user-centric [43], i.e., pro-actively maintaining the perceived quality of synthesized images according to user interactions on each connected client, while preventing

redundant depth image transmission. The transmission scheme comprises a TA and a SC, where a transmission adaptation algorithm and a dynamic negotiation mechanism cooperate to optimize depth image transmission against user perceived quality and different network bandwidth.

1) TRANSMISSION ADAPTOR

The main task of TA is fetching reference depth images according to user interactions, which are transformed into user viewpoint movement and synchronized by SC. As illustrated in Fig. 7, v_1 is current reference viewpoint whose reference depth image has already been transmitted to the client-side. When user viewpoint moves within the viewpoint interval of v_1, i.e., $[v_1, v_2]$ or $[v_1, v_0]$, no redundant reference depth image is required to transmit, while the perceived quality of synthesized images is maintained according to the quality threshold.

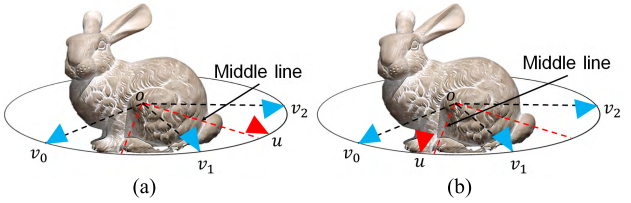

FIGURE 7. Example of transmission adaptor operations: Blue triangles indicate predicted reference viewpoints. *u* (red triangle) indicates user viewpoint. (a) and (b) illustrate the situation that user viewpoint moves across the middle line of two adjacent references, which is the timing for new reference depth image transmission. The reference depth images of v_2 and v_0 are then transmitted in the situation of (a) and (b), respectively.

In contrast, when user viewpoint is moving outside the viewpoint interval of the current reference viewpoint, TA fetches a new reference viewpoint from the predicted reference viewpoint set which is the nearest. We particularly preserve a time interval for depth image rendering and compressing. Hence, the fetching of reference viewpoint occurs when user viewpoint is moving across the middle line of current reference viewpoint interval, as illustrated in Fig. 7. To reduce possible bias of user viewpoint, a Kalman filter is additionally integrated to reinforce the prediction of user viewpoint movement, formulated as follows:

$$u[t+1] = u[t] + \vec{v}(u[t]), \tag{12}$$

where $u[t+1]$ denotes the predicted user viewpoint for a near future, $u[t]$ is the historic user viewpoint and $\vec{v}(u[t])$ indicates its current velocity. The overall work of TA is depicted in Algorithm 2.

Different from previous work, which ignore the existence of varying network bandwidth, our proposed transmission scheme is scalable to different network bandwidths. Specifically, TA adaptively changes the scale of reference viewpoint set according to the feedback from SC. For instance, it changes to a larger scale reference viewpoint set when SC notifies that the available network bandwidth is insufficient. By doing so, the fetched reference viewpoint has a larger viewpoint interval, extending the transmission timing. Suppose T_{start} and T_{end} are the start and end time, and user viewpoint moves along a virtual path with constant velocity. The depth image transmission timings are indicated with triangles as in Fig. 8.

Algorithm 2 Transmission Adaptation Algorithm

Require: $V = \{\cdots, v_{i-1}, v_i, v_{i+1}, \cdots\}$
Require: $v_{ref} = v_i$
Require: $u[t], \vec{v}(u[t])$
  $u[t+1] \leftarrow u[t] + \vec{v}(u[t])$
  if $(u[t+1] - v_i) \geq (v_{i+1} - u[t+1])$ **then**
    $v_{ref} = v_{i+1}$
  else if $(v_i - u[t+1]) \geq (u[t+1] - v_{i-1})$ **then**
    $v_{ref} = v_{i-1}$
  end if
  Rendering and compressing $< I(v_{ref}), D(v_{ref}) >$
  Transmitting $< I(v_{ref}), D(v_{ref}) >$

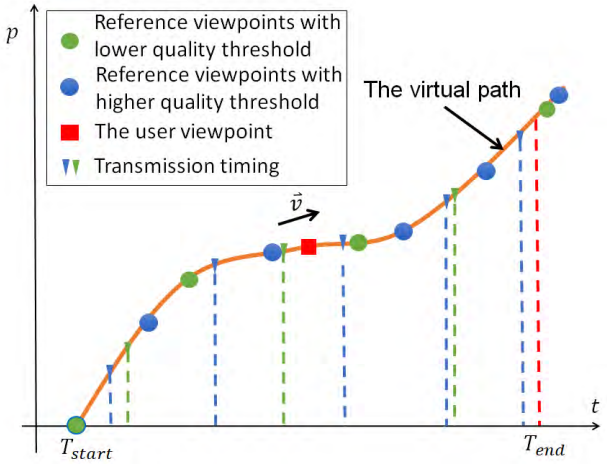

FIGURE 8. Illustration of transmission adaptation with multi-scale reference viewpoint set: Movement of user viewpoint between time T_{start} and T_{end} is illustrated. Green circles and blue circles indicate reference viewpoints from large- and small-scale reference viewpoint set, respectively. Green and blue triangles indicate the transmission timings.

Note that the blue reference viewpoints are fetched from a small-scale reference viewpoint set, and the green ones are fetched from a large-scale one. We can see in Fig. 8 that five depth image transmission occurs with respect to the blue reference viewpoints, while only three transmission occurs with respect to the green ones. The transmission frequency is reduced by 40% with the scale adaptation. The perceived quality of synthesized image is degraded due to the enlarged viewpoint interval, but still being consistent with the preset quality threshold.

2) SCALABILITY CONTROLLER

SC mainly monitors the perceived quality of synthesized images at the client-side, as well as the runtime available network bandwidth, with which to suggest TA the scale of

FIGURE 9. Visual quality of synthesized images warped under the same reference viewpoint but with reference depth images of different resolutions. (a) The undistored image with 768 × 1024 pixel. (b) Synthesized by reference depth image with 768 *times* 1024 pixel. (c) Synthesized by reference depth image with 600 *times* 800 pixel. (d) (b) Synthesized by reference depth image with 540 *times* 720 pixel.

reference viewpoint set. Our proposed NR JND-based metric is used to assist such a monitoring, and that transmitting undistorted ground truth images for quality assessment is avoided. To further reduce rendering cost and network bandwidth consumption, SC takes into account depth image resolution, notifying TA with an appropriate the proper rendering resolution of next reference depth image with perceived quality maintained.

As depicted in Fig. 9, geometric distortions are tolerable even if reference depth image resolution is changed. Images from Fig. 9(b) to Fig. 9(d) are synthesized from the same reference depth image but with different resolutions. The perceived quality of them are hardly distinguishable, especially for images of Fig. 9(b) and Fig. 9(c). Both of them exhibit similar geometric distortions.

Consequently, with the perceived quality of synthesized images being retained, there is still room for reducing the reference depth image resolution. SC is responsible to request for receiving lower-resolution reference depth images to reduce network bandwidth consumption. Algorithm 3 depicts how such a dynamic negotiation process works.

Algorithm 3 Dynamic Negotiation Algorithm

```
Initialize the rendering resolution res_display;
Initialize the smallest scale of reference viewpoint set;
for run time do
  if BW ≥ BW_available then
    if Q_syn(I'(u)) ≥ Q_allow then
      Decrease the rendering resolution
    else
      Increase the rendering resolution
    end if
  else if BW < BW_available then
    Increase the scale of reference viewpoint set
  end if
end for
```

Dynamic negotiation is initiated by SC. When a mobile device connects to the system, SC informs TA to transmit the first reference depth image with a display resolution of $res_{display}$, meanwhile the reference viewpoint set with the smallest scale is suggested. During runtime, SC measures the perceived quality of synthesized images with our NR JND-based metric, comparing it with a preset quality threshold Q_{allow}. A low rendering resolution is suggested in case if the perceived quality is maintained above Q_{allow}.

Scalability is controlled by two ways. First, SC informs TA for changing the scale of reference viewpoint set according to network bandwidth conditions, e.g., a larger-scale reference viewpoint set is preferred when the available bandwidth becomes insufficient. Second, SC can lower its quality threshold Q_{allow} to allow more rendering resolution reduction, supporting severe network bandwidth conditions.

The proposed transmission scheme is also adaptive to multiple clients. It monitors the display resolution, user viewpoint movement and available network bandwidth of each client. The adaptation algorithm and dynamic negotiation algorithm are involved to ensure the perceived quality on each mobile is optimized for these criteria.

V. EXPERIMENT RESULTS

We present our simulated settings for supporting interactive 3D graphics on mobile devices. We also study the performance of our proposed JND-based synthesized image metric. After that, the performance of our reference viewpoint prediction and adaptive transmission scheme are evaluated in Section V-C and Section V-D, respectively.

A. SIMULATION SETTINGS

We simulate the interactive 3D graphics scenarios with three 3D scene models, including *City Paris*, *Fairy Forest* and *Car*, as shown in Fig. 10. Table 2 lists their geometric complexities. These models have different characteristics, facilitating us to evaluate how well our framework performs under different situations. Specifically, *City Paris* contains complex object

(a) (b) (c)

FIGURE 10. Three testing 3D scenes: (a) City Paris, (b) Fairy Forest, (c) Car.

TABLE 2. Geometry information of the testing 3D scenes.

Scene	Vertices	Patches	File size
City Paris	1974189	658063	178 MB
Fairy Forest	97124	174117	54 MB
Car	1240887	1981300	172 MB

structures, easily being occluded from a given reference viewpoint. *Fairy Forest* comprises simple structures but complex intensity and texture variations. *Car* comprises highly precise surface details, which are sensitive to rendering resolution variations.

The server is simulated on a Dell OptiPlex with an Intel Core i5-3470 CPU and a NVIDIA GeForce GTX 650 graphics card. The display resolution of synthesized image is set to 768 × 1024. The interaction scenario considered in our simulations is 3D navigation, where user viewpoint moves in a plane with four possible directions, as listed in Table 9. Without loss of generality, user viewpoint is assumed to move with a constant velocity.

B. PERFORMANCE OF JND-BASED SYNTHESIZED IMAGE QUALITY METRIC

Performance of the proposed metric is evaluated on the IRCCyN/IVC DIBR image database [46], which is the only available public DIBR image database with subjective scores at the time of developing this work. It consists of 12 original images and their corresponding 84 synthesized images, which are generated using seven DIBR approaches. A discrete rating scale from 1 to 5 is adopted in its subjective experiment, and the subjective scores of these images in the database are provided in the form of Mean Opinion Score (MOS). To properly represent human perception, the Difference Mean Opinion Score (DMOS) is calculated from MOS values, which is re-scaled to [0, 1] in terms of our measured results.

For performance evaluation, three recognized criterions, namely Pearson Linear Correlation Coefficient (PLCC), Spearman Rank Order Correlation Coefficient (SROCC) and Root Mean Square Error (RMSE) are adopted. PLCC and RMSE are used to measure the measurement accuracy and SROCC is adopted to evaluate the monotonicity. Higher values of PLCC and SROCC and lower value of RMSE indicate better performance.

Table 3 shows the experimental results with the IRCCyN/IVC DIBR database. For comparison, we choose

TABLE 3. Comparison of our and other metrics with IRCCyN/IVC DIBR image dataset. PLCC, SROCC and RMSE are indicators evaluating metric performance with subjective values. The best two performers under each indicator are highlighted.

	Type	PLCC	SROCC	RMSE
MSE	FR, 2D	0.4279	0.4610	0.6018
SSIM [32]	FR, 2D	0.3703	0.3069	0.6185
3DSwIM [36]	FR, DIBR	**0.6623**	**0.6158**	0.4988
Ours	FR, DIBR	**0.5730**	**0.5654**	**0.4351**
Ours	NR, DIBR	0.5140	0.5228	**0.4790**

conventional 2D image metrics, including MSE and SSIM [32] and DIBR metrics 3DSwIM [36]. MSE is based on pixel errors, while SSIM is sensitive to structures. 3DSwIM measures DIBR synthesized image with multiple Nature Scene Statistics (NSS) priors in wavelet domain. Despite recent metrics, like MW-PSNR [37] and MP-PSNR [38], achieve better performance, they introduce time-consuming operations, e.g., wavelet decomposition, morphological operation, and that are inadequate for real-time remote rendering.

Observing from Table 3, conventional 2D image quality metrics are not effective in evaluating the quality of synthesized images. Their best PLCC result is below 0.5 and maximum SROCC is only 0.4610. These results imply both pixel errors (MSE) and structure distortions (SSIM) are not suitable for measuring geometric distortions in synthesized image. Comparatively, 3DSwIM produces much better values in PLCC and SROCC. In contrast, our FR metric achieves the best performance on RMSE, meanwhile our NR metric achieves comparable performance. More recent DIBR-related metrics [37]–[41] follow the design of 3DSwIM, integrating more complex NSS priors with wavelet transformation, multi-scale representations or auto-regression. Despite they offer better performance than our metrics, the domain transformations or multi-scale representations are time-consuming for real-time computation on mobile devices, which is critical for supporting interactive user interactions.

To verify the computational efficiency of our work, we compare time spent on assessing the quality of a depth image with different methods, as shown in Table 4. Note that the CPU time cost is tested with MATLAB, while the GPU time cost is obtained with C++.

TABLE 4. Time efficiency of our JND-based metric on PC with 768 × 1024 depth image resolution.

	CPU (ms)	GPU (ms)
MSE	237	13.4
SSIM	524	13.8
3DSwIM	7277	/
Ours (FR)	624	15.3

We can see from Table 4 that our metric achieves comparable time efficiency with MSE and SSIM, performing

10 times faster than 3DSwIM. The efficiency bottleneck of 3DSwIM is wavelet transform, while our metric can be easily paralleled on GPUs.

We additionally evaluate our FR and NR metric on 40 synthesized images from our three tested 3D scene models. The scatter plots between the FR metric predicted quality scores and those of the NR metric are shown in Fig. 11. It is observed that the NR metric predicted results are highly correlated with the FR metric predicted ones, having a correlation coefficient $R^2 = 0.8903$. This implies it is appropriate to use NR JND-based metric as an alternative assessing synthesized image quality.

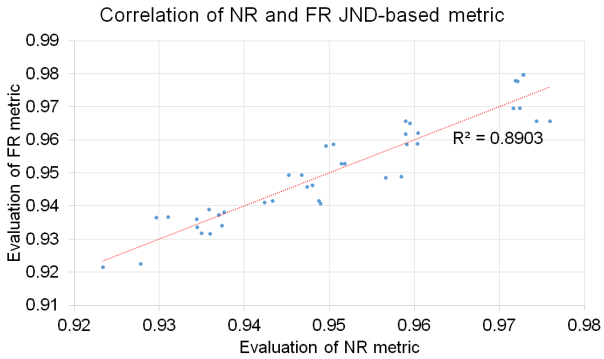

FIGURE 11. Correlation between NR and FR versions of our proposed JND-based synthesized image metric.

C. PERFORMANCE OF REFERENCE VIEWPOINT PREDICTION

We simulate reference viewpoint prediction on the three testing 3D scene models. The simulated results on *City Paris* are depicted in details as follows. We start the full-search algorithm with a random initial viewpoint. User viewpoint is restricted to the horizontal path movement for convenience. Without loss of generality, we repeat the experiment five times with different initial viewpoints, denoting as $\{v_0^{(1)}, v_0^{(2)}, v_0^{(3)}, v_0^{(4)}, v_0^{(5)}\}$. We preset Q_{JND} to 0.990 and the maximal distance d_{MAX} to 100. Table 5 shows the number of predicted reference viewpoints.

TABLE 5. The number of predicted reference viewpoints. The full-search algorithm uses the quality threshold $Q_{JND} = 0.990$, where Shi *et al.* [3] with $Q_{MSE} = 224.0$ is used for comparison.

Initial Viewpoint	Ours	Shi [3]
$v_0^{(1)}$	17	20
$v_0^{(2)}$	15	18
$v_0^{(3)}$	17	19
$v_0^{(4)}$	19	23
$v_0^{(5)}$	15	19
Average Viewpoint Interval	6.02	5.05

As a comparison, we simulate the above experiment using $Q_{MSE} = 224.0$, which is used in [3]. Since MSE ranges from 0 to $+\infty$, it is hardly comparable with our JND-based metric. We have tested 30 synthesized images from *CityParis*, obtaining their MSE results and our measured results. It is observed that MSE with values between 172.0 to 224.0 are corresponding to $Q_{JND} = 0.990$. We thereby choose $Q_{MSE} = 224.0$, for it is the maximal value that produces the sparsest reference viewpoints. As shown in Table 5, our predicted reference viewpoints are still less redundant than that produced by MSE, i.e., the average viewpoint interval with our proposed JND-based metric is larger than that with Shi *et al.* [3].

For the convenience of analysis, we denote candidate viewpoints according to their distances to the initial viewpoint, e.g., v_{46} is the viewpoint with a distance of $46 \times \Delta d$ from v_0. The subscript difference hence indicates the viewpoint interval.

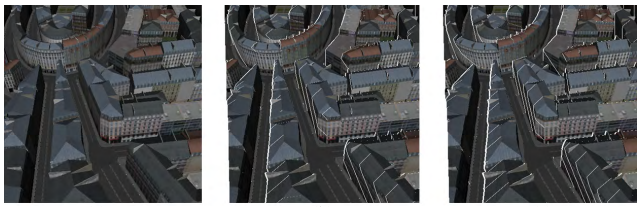

FIGURE 12. Visual quality of synthesized images (Blank pixels indicate geometric distortions): (a) is the undistorted image of viewpoint v_{54}. (b) and (c) are synthesized images of v_{54} constructed from reference depth images v_{52} and v_{46}, respectively. Although the two synthesized images have different pixel errors, their perceived qualities are consistent.

Fig. 12 shows the synthesized image of v_{54} from *City Paris* with reference viewpoint v_{52} predicted by MSE and v_{46} predicted by our JND-based metric. In Fig. 12, the perceived quality of the two synthesized images is hardly distinguished. The image quality assessment results further validate our perception, where the two synthesized images have the same results measured by our JND-based metric. However, the MSE results vary significantly, increasing from 172.0 to 224.0. Therefore, our predicted reference viewpoint v_{46} has a $8 \times \Delta d$ viewpoint interval, which is larger than of MSE predicted reference viewpoint v_{52}, which is only $2 \times \Delta d$. More results are shown in Fig. 13(b) and Fig. 13(c). The target viewpoint and the predicted reference viewpoints by MSE and our JND-based metric in Fig. 13 are listed in Table 6. From Fig. 13 and Table 6, we can see that our JND-based metric predicts the reference viewpoint with larger viewpoint intervals in general, while well maintaining perceived quality.

We further explore the perceived quality of synthesized images on different scales of reference viewpoint set. A three-scale reference viewpoint set is constructed for each testing 3D scene model with $Q_{JND} = 0.990, 0.985$ and 0.980, respectively. Fig. 13 depicts synthesized images with reference viewpoints under different scales. Take *City Paris* as an example, user viewpoint is v_{54}, where the predicted reference viewpoints are v_{46}, v_{45} and v_{43} in terms of different scales. We can conclude from the figure that the perceived quality of

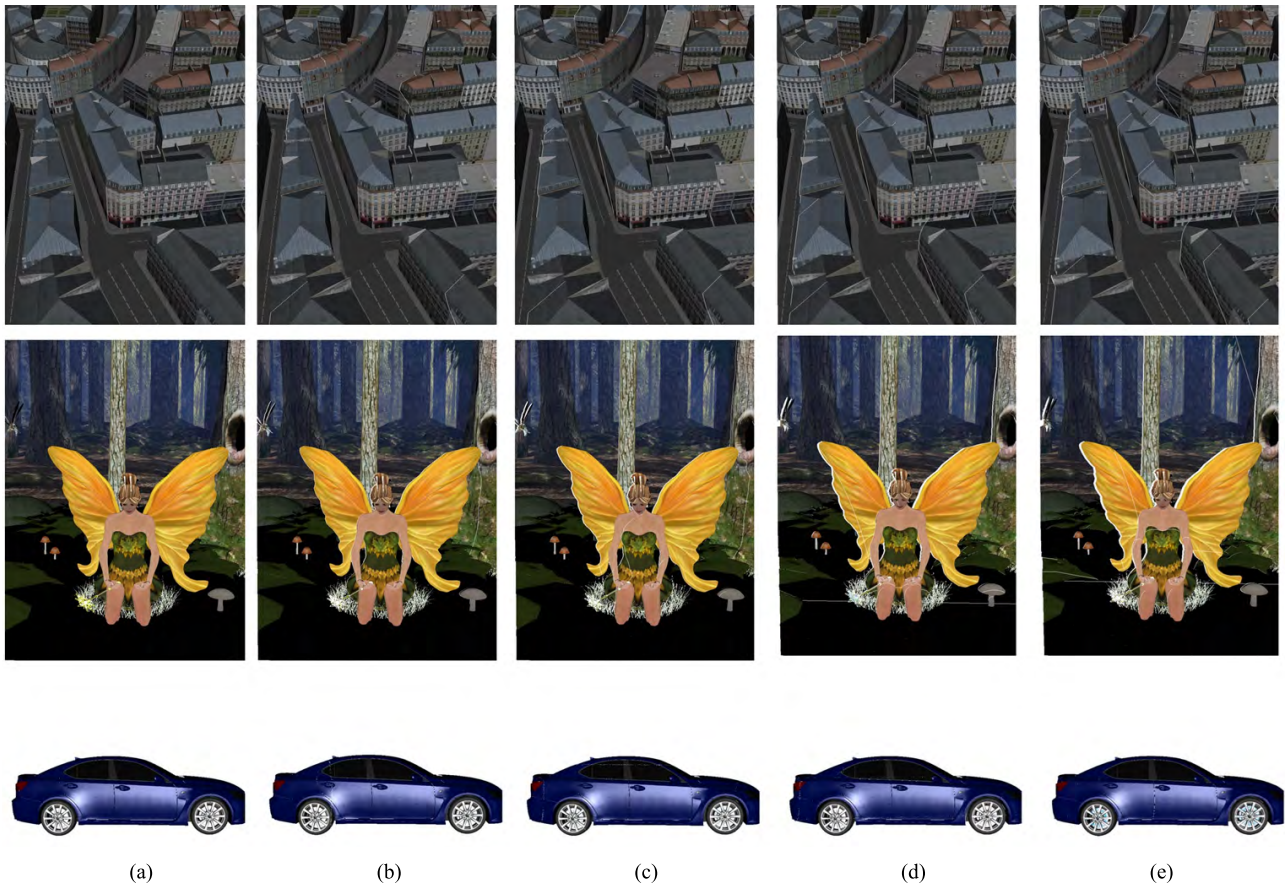

FIGURE 13. Synthesized images with different reference viewpoints predicted by MSE and our JND-based metric. (a) The undistored image. (b) Synthesized with MSE predicted reference view point. (c) Synthesized with our predicted reference view point ($Q_{thres} = 0.990$). (d) Synthesized with our predicted reference view point ($Q_{thres} = 0.985$). (e) Synthesized with our predicted reference view point ($Q_{thres} = 0.980$).

TABLE 6. The target viewpoint u and predicted reference viewpoints v_{ref} by MSE and our JND-based metric (with different quality thresholds), in corresponding to Fig. 13. The number in each bracket indicates the viewpoint interval.

3D Contents	u	v_{ref}			
		MSE [3]	$Q_{JND} = 0.990$	$Q_{JND} = 0.985$	$Q_{JND} = 0.980$
City Paris	v_{54}	$v_{52}(2)$	$v_{46}(8)$	$v_{45}(9)$	$v_{43}(11)$
Fairy Forest	v_{39}	$v_{41}(2)$	$v_{45}(6)$	$v_{47}(8)$	$v_{49}(10)$
Car	v_{48}	$v_{45}(3)$	$v_{43}(5)$	$v_{42}(6)$	$v_{40}(8)$

TABLE 7. Comparison of transmission frequency.

Method	Frames	Description
Mark [1]	≥ 50	Constantly transmit 5 frames per second.
Bao [6]	≥ 50	The same as [1], but transmitting differential depth image instead.
Shi [3]	$12 \times n$	Reference viewpoint is predicted by MSE, n indicates all possible directions of user viewpoint movement.
Ours	10	Reference viewpoint predicted with $Q_{JND} = 0.990$
Ours	8	Reference viewpoint predicted with $Q_{JND} = 0.985$
Ours	6	Reference viewpoint predicted with $Q_{JND} = 0.980$

the synthesized images slightly degrades by reducing Q_{JND}. However, the viewpoint interval is then enlarged, reducing the transmission frequency as depicted in Fig. 8.

D. PERFORMANCE OF ADAPTIVE TRANSMISSION MECHANISM

We further demonstrate the performance of our adaptive transmission mechanism. We still simulate the virtual viewpoint moving along the horizontal path with a constant velocity. Transmission of a reference depth image occurs when user viewpoint moves across the middle line of current reference viewpoint interval, as illustrated in Fig. 7. Table 7 lists the total transmitted reference depth images within 10 seconds. To evaluate the efficiency of our transmission adaptation algorithm, we examine three typical DIBR-based remote rendering framework, including [1], [6], and [3], analyzing their transmission frequency against our method. Observing from Table 7, our method transmits the least number of reference depth images during the same

TABLE 8. Rendering resolutions of reference depth images transferred to client-side with the proposed dynamic negotiation algorithm, where different Q_{allow} are also evaluated.

Q_{allow}	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	Data reduction
0.90	1024×768	1024×768	800×600	800×600	800×600	800×600	800×600	1024×768	32.0%
0.85	1024×768	800×600	720×540	512×384	512×384	512×384	512×384	720×540	55.0%
0.80	1024×768	800×600	720×540	512×384	400×300	400×300	400×300	512×384	61.7%

time interval. As mentioned before, [1] transmits one reference depth image for every 200ms, thereby inducing 50 transmitted frames in total. Reference [6] follows the same transmission strategy but reduces data size of each frame to some extent. [3] implements a similar transmission strategy as ours, but predicts reference viewpoint by MSE. Its average viewpoint interval is shorter than ours, inducing redundant depth image transmission. Moreover, [3] transmits n reference depth images each time, where n indicates all possible paths of user viewpoint movement. We however transmit only one reference depth image each time, where the prediction bias of user viewpoint is reduced by the Kalman filter.

We additionally evaluate the effect of our dynamical negotiation algorithm by setting $Q_{allow} = 0.90$ and $res_{display} = 1024 \times 768$, recording the rendering resolution of the transmitted reference depth images (with $Q_{JND} = 0.985$). An ideal network bandwidth is assumed. As seen from Table 8, the rendering resolution varies among different reference depth images, thereby reducing transmitted data by at least 32.0% (with $Q_{allow} = 0.90$). The transmission reduction further benefits from decreased Q_{allow}. We can see from Table 8 that the total transmitted data is reduced by 55.0% and 61.7%, respectively.

VI. PROTOTYPE AND SYSTEM EVALUATION

We now study the performance of our scalable remote rendering framework with the scenarios of interactive 3D graphics on mobile devices. A prototype system is presented in Section VI-A. System scalability in terms of multiple clients is evaluated in Section VI-B.

A. PROTOTYPE IMPLEMENTATION

The server-side is implemented on a Dell OptiPlex with an Intel Core i5-3470 CPU and a NVIDIA GeForce GTX 650 graphics card. The client-side are running on several low-end mobile devices, with configures of LG Nexus 4 with a Qualcomm Adreno 320 Graphics card and 1280×768 screen resolution, and HTC One (M8) with a Qualcomm Adreno 330 Graphics card and 1920×1080 screen resolution. A Wi-Fi connection with 11Mbps maximum bandwidth is served for depth image transmission.

1) SERVER-SIDE

TA is running on the server-side. Note that the multi-scale reference viewpoint set is pre-determined in terms of 3D contents. Besides, the server-side contains a *rendering engine* and a *depth image encoder*. For the rendering engine, we choose an open-source C++ library *OpenSceneGraph* (OSG). The 3D source contents are organized and rendered independently for different clients. Given a reference viewpoint, the texture image is directly rendered, with the Z-buffer transformed into the depth map. Our depth image encoder is depth-independent. We simply encode the texture image with an open-source library *JPEG2000*, while for the depth map, we adapt the down-sampling framework proposed in [14]. Particularly, the depth map is down-sampled with a guidance of mesh saliency, forming a sparse representation. The compressed texture image and depth map are finally streamed to the client concurrently.

2) CLIENT-SIDE

SC is running on the client-side. A *local renderer* and a *depth image decoder* is also implemented in corresponding to the server-side. Since the most complex rendering task is handled on server-side, we just design a lightweight local renderer with *OpenGL ES*. It renders synthesized images using 3D warping, which is a double warping strategy [1]. It first warps two reference depth images to a user viewpoint, and then blends the two produced synthesized images into one, which can reduce most geometric distortions. Besides, we maintain a depth buffer on the client-side, restoring the recently used reference depth images. When the network is blocked or the predicted user viewpoint is wrong, we can use the standby reference depth images instead. The depth image decoder decompresses the texture image with *JPEG2000*, while reconstructing depth map by an edge diffusion algorithm [14].

3) USER INTERACTION INTERFACE

We support finger touch on the screen of a mobile device. The touch actions are continuously monitored and transformed into user viewpoint movement in the world coordinates of the 3D scene model. Specifically, our manipulator supports three interactive 3D graphics scenarios, including *Model browsing*, *3D Navigation* and *Virtual Environment*. In the scenarios of model browsing, user can watch a 3D scene model from different viewpoints distributing on a trackball. 3D navigation simulates user observation of a city with the bird-eye. User viewpoint travels in a plane and looks down from above. For virtual environment walkthrough, a first-person walking action is supported, where user viewpoint movement is driven by virtual head motion. Table 9 lists the details of supported user viewpoint movements of different interaction patterns.

TABLE 9. Supported interaction patterns and the corresponding user viewpoint movements in our prototype.

Interaction pattern	Supported virtual viewpoint movement
Model Browsing	right-rotation, left-rotation, scale-up, scale-down
3D Navigation	left-toward, right-toward, head-toward, back-toward
Virtual Environment Walkthrough	right-rotation, left-rotation, moving-forward, moving-backward

B. SYSTEM EVALUATION

With the prototype system, we evaluate our proposed scalable remote rendering framework from two aspects. First, we study the performance of system scalability with multiple clients, and then analyze its time efficiency in terms of user interaction.

1) SYSTEM SCALABILITY

To evaluate system scalability, we run the prototype system on the server-side, testing its maximum supported concurrent mobile devices. We separately choose Nexus 4 and HTC One (M8) as client-sides. Three interactive scenarios, including model browsing of *Car*, 3D navigation of *City Paris* and virtual environment walkthrough of *City Paris*, are tested on each mobile device, respectively. Without loss of generality, each scenario is tested five times. Finally, we average the transmission frequency, actual consumed network bandwidth and the maximum supported clients, as shown in Table 10. The three quality thresholds Q_{JND} correspond to the three scales of reference viewpoint set. BW_{ava} is the allocated bandwidth for each mobile device. f_{tran} is the average depth transmission frequency, and BW_{avg} is the actual consumed network bandwidth. *Clients* indicates the maximum supported clients under Wi-Fi bandwidth constraint.

TABLE 10. Scalability performance of our prototype system running on different mobile devices. LG Nexus 4 has the display resolution of 768 × 1024, and that of HTC One (M8) is 1080 × 1920.

	Q_{JND}	BW_{ava}	f_{tran}	BW_{avg}	*Clients*
Nexus 4	0.985	1.0Mb	5 fps	733 Kbps	15
	0.985	1.5 Mb	5 fps	786 Kbps	13
	0.990	1.5 Mb	8 fps	1.22 Mbps	9
HTC	0.985	1.0Mb	5 fps	710 Kbps	15
	0.985	1.5 Mb	5 fps	1000 Kbps	11
	0.990	1.5 Mb	8 fps	1.38 Mbps	7

From Table 10, we can see that our prototype system keeps a low depth image transmission frequency and network bandwidth consumption. Even for HTC One (M8) that requires a 1080 × 1920 display resolution, its transmission frequency is less than 10 fps, while the average consumed bandwidth is less than 1.5 Mbps. We can also conclude from Table 10 that our proposed multi-scale reference viewpoint prediction is efficient for improving system scalability. As shown in Table 10, the depth image transmission frequency on Nexus 4 is reduced from 8 fps to 5 fps with Q_{JND} decreasing from 0.990 to 0.985, thereby reducing actual consumed bandwidth by 35.6%. It is also shown in Table 10 that our adaptive transmission scheme effectively reduces data transmission within the same reference viewpoint set. Take Nexus 4 again as an example, the average consumed bandwidth is reduced from 786 Kbps to 733 Kbps with the same reference viewpoints (with $Q_{JND} = 0.985$ is maintained).

In summary, our prototype system is scalable to multiple concurrent mobile devices. The proposed reference viewpoint prediction and transmission scheme make the remote rendering adaptive to available network bandwidth, while the proposed JND-based metric ensures the perceived quality on the client-side. Screen shots of the three interactive scenarios on a HTC One (M8) are shown in Fig. 14, respectively.

2) TIME EFFICIENCY

We further evaluate the time efficiency by showing a detailed breakdown of procedure timing on a Nexus 4, as depicted in Table 11. Given the next reference viewpoint is determined, the server will then render the associated depth image. The rendering time T_{ren} is related to the server's computation power, e.g., costs 30ms on our server. After rendering, the texture image and depth map are compressed concurrently. Specifically, the texture image is encoded within 20ms, while the depth map is down-sampled with 141.0 to 206.0ms, based on different resolutions. We maintain two threads that separately handling the texture image and the depth map. Therefore, the compression time T_{enc} is determined by depth down-sampling. The total time cost on server-side is then denoted as:

$$T_{server} = T_{ren} + T_{enc}, \tag{13}$$

ranging from 171.0 to 236.0ms.

TABLE 11. Time efficiency of the remote rendering system under different display resolutions on a Nexus 4 device. The average timings of each step are measured in millisecond. T_{ren}, T_{enc}, T_{dec}, T_{warp}, T_{eval} and T_{rtt} denote time cost of reference depth image rendering, depth image compression, depth image de-compression, 3D warping, synthesized image quality evaluation and transmission latency, respectively. The resolution XGA, SVGA and XGA are abbreviations of 640 × 480, 800 × 600, 1024 × 768, respectively.

Resolution	T_{Server}		T_{Client}			T_{rtt}
	T_{ren}	T_{enc}	T_{dec}	T_{warp}	T_{eval}	
VGA	30.0	141.0	14.7	30.0	17.7	10.0
SVGA	30.0	191.0	20.3	30.0	22.9	15.3
XGA	30.0	206.0	24.6	41.0	28.9	27.0

The procedures running on client-side include depth image de-compression, 3D warping and synthesized image quality assessment. De-compression of texture image and depth map are parallelized. Decoding of texture image costs about 30ms, while decoding of depth map needs 20.5 to 41.0ms, since the edge diffusion algorithm is accelerated on mobile GPUs. Therefore, the de-compression time T_{dec} ranges from 30ms to 41.0ms. For low reference depth image resolution, the total de-compression time is determined by texture

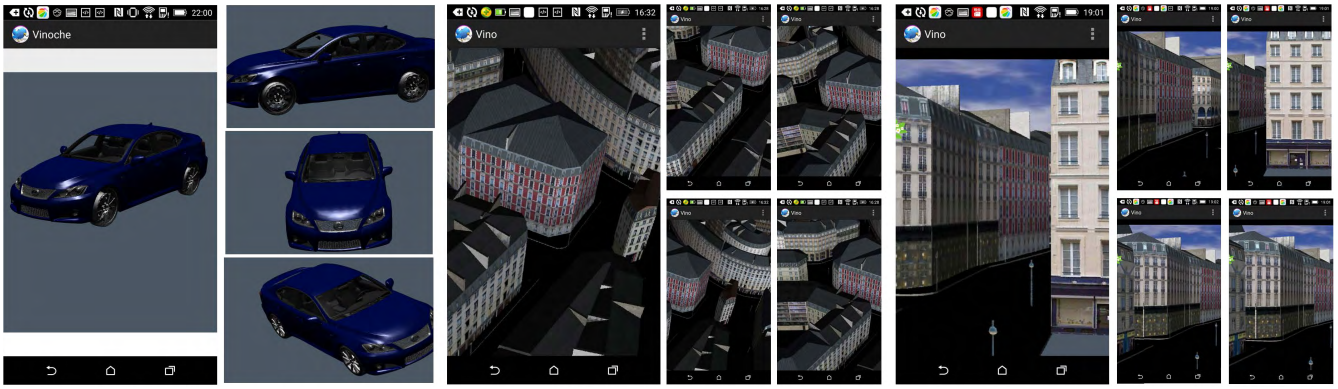

FIGURE 14. Screen shots of interactive scenarios on a HTC One (M8). (a) Model browsing car. (b) 3D navigation city paris. (c) Virtual environment walkthrough city paris.

image decoding. With increasing in resolution, depth map reconstruction becomes dominate. 3D warping T_{warp} costs 17.7 to 28.9ms. Our NR JND-based synthesized image quality assessment T_{eval} takes 10.0 to 27.0ms. However, the time cost of image quality assessment is excluded when counting interaction latency, since it is running in parallel with 3D warping and separately handled by SC. The total time cost on client-side can then be formulated as:

$$T_{client} = T_{dec} + T_{warp}, \tag{14}$$

ranging from 47.7 to 69.9ms. The interaction latency also includes round-trip time T_{rtt}, which is around 14.7-24.6ms in our evaluation environment. The total interaction latency therefore ranges from 225.7 to 328.6ms.

The interaction latency is further optimized by two means. First, we render the reference depth image in advance, i.e., when user viewpoint is moving across the middle line of current reference viewpoint interval, as addressed in Section IV-B.1. Second, historical reference depth images are cached on client-side, which can be used for synthesizing in case that new reference depth image are not available. In practice, our prototype system achieves about 200ms interaction latency, being adequate for interactive 3D graphics applications.

VII. CONCLUSION

In this paper, we have presented a scalable DIBR-based remote rendering framework based on synthesized image quality assessment. Our work puts efforts on improving depth image transmission while maintaining perceived quality on concurrent multiple clients running on mobile devices. To our knowledge, we are the first to propose a JND-based synthesized image quality metric, adopting it to reference viewpoint prediction and transmission control. Our method accounts for both perceived quality and network bandwidth adaptation. It surpasses previous works by three aspects. First, we measure the perceived quality of synthesized images with a JND-based metric, which is more consistent with human perception than pixel errors used in previous methods. Second, we predict a multi-scale reference viewpoint set to prevent redundant depth image transmission, better adapting available network bandwidth. Finally, we design an adaptive transmission scheme which is aware of both perceived quality and network bandwidth. We further consider the rendering resolution of reference depth image for saving bandwidth consumption.

The proposed remote rendering framework is still improvable. We mainly focus on depth image transmission in this paper, but leaving optimizations for other components in a remote rendering framework. Take depth map compression as an example, saliency-guided down-sampling preserves geometric details in synthesized images, but consumes too much time. Besides, we would like to adopt user interaction habits assisting user viewpoint prediction, in order to provide more smooth quality of experience for user interaction.

REFERENCES

[1] W. Mark, "Post-rendering 3D image warping: Visibility, reconstruction, and performance for depth-image warping," Chapel Hill, NC, USA, Tech. Rep., 1999.
[2] A. Boukerche and R. W. N. Pazzi, "Scheduling and buffering mechanisms for remote rendering streaming in virtual walkthrough class of applications," in *Proc. 2nd ACM Int. Workshop Wireless Multimedia Netw. Perform. Modeling*, 2006, pp. 53–60.
[3] S. Shi, K. Nahrstedt, and R. Campbell, "A real-time remote rendering system for interactive mobile graphics," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 8, no. 3s, Oct. 2012, Art. no. 46.
[4] S. Zellmann, M. Aumüller, and U. Lang, "Image-based remote real-time volume rendering: Decoupling rendering from view point updates," in *Proc. ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Aug. 2012, pp. 1385–1394.
[5] P. Bao and D. Gourlay, "Low bandwidth remote rendering using 3D image warping," in *Proc. Int. Conf. Vis. Inf. Eng. (VIE)*, Jul. 2003, pp. 61–64.
[6] P. Bao and D. Gourlay, "A framework for remote rendering of 3-D scenes on limited mobile devices," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 382–389, Apr. 2006.
[7] D. Pająk, R. Herzog, E. Eisemann, K. Myszkowski, and H.-P. Seidel, "Scalable remote rendering with depth and motion-flow augmented streaming," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 415–424, 2011.
[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[9] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[10] B.-B. Chai, S. Sethuraman, and H. S. Sawhney, "A depth map representation for real-time transmission and view-based rendering of a dynamic 3D scene," in *Proc. 1st Int. Symp. 3D Data Process. Vis. Transmiss.*, Jun. 2002, pp. 107–114.

[11] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li, "Kinect-like depth data compression," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1340–1352, Oct. 2013.

[12] V.-A. Nguyen, D. Min, and M. N. Do, "Efficient techniques for depth video compression using weighted mode filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 189–202, Feb. 2013.

[13] D. Pająk *et al.*, "Perceptual depth compression for stereo applications," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 195–204, 2014.

[14] M. Gu *et al.*, "Saliency-driven depth compression for 3D image warping," in *Pacific Graphics Short Papers*. 2014.

[15] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," *Proc. SPIE*, vol. 6055, pp. 93–100, Jan. 2006.

[16] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[17] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 65–72, 2009.

[18] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3D video systems," *EURASIP J. Image Video Process.*, vol. 2008, Dec. 2009, Art. no. 438148.

[19] P. Ndjiki-Nya *et al.*, "Depth image based rendering with advanced texture synthesis," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Singapore, Jul. 2010, pp. 424–429.

[20] M. Köppel *et al.*, "Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering," in *Proc. 17th IEEE Int. Conf. Image Process.*, Hong Kong, Sep. 2010, pp. 1809–1812.

[21] C. Zhu and S. Li, "Depth image based view synthesis: New insights and perspectives on hole generation and filling," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 82–93, Mar. 2016.

[22] M. Schmeing and X. Jiang, "Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2160–2173, Dec. 2015.

[23] S. Li, C. Zhu, and M.-T. Sun, "Hole filling with multiple reference views in DIBR view synthesis," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2018.2791810.

[24] F. W. B. Li, R. W. H. Lau, D. Kilis, and L. W. F. Li, "Game-on-demand: An online game engine based on geometry streaming," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 7, no. 3, Aug. 2011, Art. no. 19.

[25] H.-Y. Shum, *Image-Based Rendering*. Springer, 2008.

[26] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.

[27] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.

[28] S. Shi, M. Kamali, K. Nahrstedt, J. C. Hart, and R. H. Campbell, "A high-quality low-delay remote rendering system for 3D video," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 601–610.

[29] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, "Layered depth images," in *Proc. 25th Annu. Conf. Comput. Graph. Interact. Techn.*, 1998, pp. 231–242.

[30] Z. Jiang *et al.*, "PanoWalk: A remote image-based rendering system for mobile devices," in *Advances in Multimedia Information Processing—PCM*. Springer, 2006, pp. 641–649.

[31] C. Zhang and T. Chen, "Nonuniform sampling of image-based rendering data with the position-interval-error (PIE) function," *Proc. SPIE*, vol. 5150, pp. 1347–1358, Jun. 2003.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] C. D. Regis *et al.*, "Video quality assessment based on the effect of the estimation of the spatial perceptual information," in *Proc. 30th Brazilian Symp. Telecommun. (SBrT)*, 2012.

[34] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[35] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[36] F. Battisti, E. Bosc, M. Carli, P. Le Callet, and S. Perugia, "Objective image quality assessment of 3D synthesized views," *Signal Process., Image Commun.*, vol. 30, pp. 78–88, Jan. 2015.

[37] D. Sandić-Stanković, D. Kukolj, and P. Le Callet, "DIBR synthesized image quality assessment based on morphological wavelets," in *Proc. 7th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, May 2015, pp. 1–6.

[38] D. Sandić-Stanković, D. Kukolj, and P. Le Callet, "DIBR-synthesized image quality assessment based on morphological multi-scale approach," *EURASIP J. Image Video Process.*, vol. 2017, Jul. 2016, Art. no. 4.

[39] Y. Zhou, L. Li, K. Gu, Y. Fang, and W. Lin, "Quality assessment of 3D synthesized images via disoccluded region discovery," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 1012–1016.

[40] K. Gu, V. Jakhetiya, J.-F. Qiao, X. Li, W. Lin, and D. Thalmann, "Model-based referenceless quality metric of 3D synthesized images using local image description," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 394–405, Jan. 2018.

[41] S. Tian, L. Zhang, L. Morin, and O. Déforges, "NIQSV+: A no-reference synthesized view quality assessment metric," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1652–1664, Aug. 2018.

[42] C.-H. Chou and Y.-C. Li, "A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 6, pp. 467–476, Dec. 1995.

[43] H. R. Wu, "Introduction: State of the play and challenges of visual quality assessment," in *Visual Signal Quality Assessment*. Cham, Switzerland: Springer, 2015, pp. 1–30.

[44] X. K. Yang, W. S. Ling, Z. K. Lu, E. P. Ong, and S. S. Yao, "Just noticeable distortion model and its applications in video coding," *Signal Process., Image Commun.*, vol. 20, no. 7, pp. 662–680, Aug. 2005.

[45] A. Liu, W. Lin, M. Paul, C. Deng, and F. Zhang, "Just noticeable difference for images with decomposition model for separating edge and textured regions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1648–1652, Nov. 2010.

[46] E. Bosc, "Towards a new quality metric for 3-D synthesized view assessment," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 7, pp. 1332–1343, Nov. 2011.

XIAOCHUAN WANG received the M.Sc. degree from Beihang University, Beijing, China, in 2012, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Virtual Reality Technology and System. His current research interests include mobile graphics, remote rendering, image quality assessment, and multi-view video systems.

XIAOHUI LIANG (M'87) received the Ph.D. degree in computer science and engineering from Beihang University in 2002. He is currently a Professor with the State Key Laboratory of Virtual Reality Technology and System, Beihang University. His research interests include computer graphics, animation, visualization, and virtual reality.

BAILIN YANG received the Ph.D. degree from the Department of Computer Science, Zhejiang University, in 2007. He is currently a Professor with the Department of Computer and Electronic Engineering, Zhejiang Gongshang University. His research interests are in mobile graphics, real-time rendering, and mobile game.

FREDERICK W. B. LI received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2001. He is currently an Assistant Professor with Durham University, Durham, U.K. Before the appointment, he was an Assistant Professor at The Hong Kong Polytechnic University and the Project Manager of a Hong Kong Government Innovation and Technology Fund funded project. His current research interests include distributed virtual environments, computer graphics, and e-learning systems. He has served on the Conference Committee of a number of conferences, including the Program Co-Chair of ICWL from 2007 to 2008, in 2013, and in 2015, the IDET from 2008 to 2009, and the Workshop Co-Chair of ICWL 2009 and U-Media 2009. He has served as a Guest Editor of some special issues for the *International Journal of Distance Education Technologies* and the *Journal of Multimedia*.